RTES Technical Status and Future R&D

BTeV Document 1219

Jim Kowalkowski, Paul Sheldon

1 Introduction

This Real Time Embedded Systems project (RTES) is a collaborative effort between computer scientists and high energy physicists. It is researching the design and implementation of high-performance, heterogeneous, fault-tolerant and fault-adaptive real-time embedded systems. The specific application that will drive this research and provide a test platform for it is the trigger and data acquisition system for BTeV. The investigators consist of Computer Engineers and Scientists who are experts in embedded systems, real-time systems, and fault tolerant computing, and the leaders of the BTeV trigger and data acquisition system development team. As a part of its R&D program, the team will implement a series of systems of increasing size and complexity, using the experience gained at each stage to refine and improve its tools and methodologies. Each of the groups involved in this project have relevant experience and a mix of existing software and hardware that they bring into the project. The RTES project is funded through the NSD ITR program, NSF Award #ACI-0121658.

2 Research and development summary

The RTES group has built DSP-based hardware to emulate the BTeV prototype trigger environment. This hardware is being used to learn about the limitations and capabilities of DSPs and also to design and create the fault handling software that will be used in the BTeV trigger. The TI 6711 DSPs are the same ones that the Pixel trigger group is using in their pre-prototype farmlet.

The Vanderbilt group has created two types of boards. The first is a PCI card capable of holding up to three DSPs interconnected by an FPGA for use inside a developer's PC. The second is a VME card that holds up to four daughter cards, each this a FPGA front-end for interprocessor communications. Each university group has received their own PCI board with two DSPs for prototyping and testing. There is a VME crate at Vanderbilt with 16 cards for a total of about 64

processors that will be used as a mini-farm for testing the scalability of the software. This system has the capability of emulating or simulating both hardware and software faults that will appear in the time critical, embedded processor portion of the real level 1 trigger. The Vanderbilt group has also made Windows-based host software and a simple DSP software kernel available for use by RTES for downloading and running code.

The University of Illinois group (UIUC) has been concentrating on analyzing various aspects of the BTeV environment in order understand how ARMORs (a software framework and components for creating reliable applications) will be used to solve the problem and what enhancements will be necessary.

The Pittsburgh and Syracuse groups are defining the concept of a VLA (Very Lightweight Agent) and writing specification for them. Syracuse has begun experimenting with ideas to see what can be accomplished on a DSP. The VLAs will be responsible for capturing faults and making simple decisions very close to the application processes (likely tightly coupled to it) in a very restricted and confined environment.

The Vanderbilt group, in addition to supplying the core boot kernel for the prototype, is working on applying their mature modeling methodologies to the prototype and trigger. They are integrating BTeV trigger concepts and constructs into their existing GME (Generic Modeling Environment) tools. The tools can be used for graphically generating fault handling executables and configuration scripts. RTES is currently using these tools to put together programs and configuration to operate the PCI-based DSP boards.

The Syracuse group has begun defining job submission and processing simulations to studying scheduling algorithms. The research is aimed at defining a scheduling that will increase the utilization and throughput of user submitted analysis or reconstruction jobs on a farm.

3 Relationship to the BTeV trigger

The research and development efforts from the RTES group will be used to help satisfy several of the trigger requirements listed in BTeV document 878.

- *2.6-2 (Uptime)*: The hierarchical nature of the fault reporting and mitigation system will allow for quick response to problems. Automated recovery and problem prediction procedures will minimize downtime.
- *2.9-1 (Component failure)*: ARMORs and VLAs will be responsible for taking components out of service and bring them back. They will contain the rules on reconfiguration and how to compensate for lose of resources.
- *2.9-2 (Trigger arbitration)*: The ARMORs and VLAs will be responsible for capturing the rate information, carrying out corrective actions, and delivering configuration information to the trigger application.
- *2.9-3 (Purging):* The notifications lost data will be captured and records by RTES components.
- *2.9-4 (Timeouts):* The RTES components will be used to determine and cause timeouts and also record and monitor this information.

- *2.11-1 (Control and monitoring)*: The RTES components will be collecting all the errors and monitoring information. These components will be entities reporting to the controls and monitoring system.
- *4.1-5 (RTES)*: Using 10% or less resources required that the RTES software be efficient in what it does and have the ability to change its usage levels depending on the resource needs of the application.

RTES subsystems will need to cooperate with other subsystems developed by BTeV in order to meet some of the requirements. Some of research efforts of RTES are focused on meeting the following requirements in regards to fault management.

- *2.7-2 (Scalability)*: An important focus of RTES research is scalability in terms of number of CPUs and running applications. The research will insure that RTES will not be the limiting factor.
- *2.11-2 (Read-back)*: The source of errors and perhaps configuration information will likely be RTES. The controls and monitoring system will need to communicate with RTES.
- *2.11-3 (Initialization)*: The controls and monitoring system will likely need to send commands to RTES or through RTES for initialization, since RTES may manage many of the configuration facilities and execution environment.
- *2.12-2 (Testing)*: RTES will understand if components are operating correctly and how they fail.

4 Personal

Jeff	Appel	Fermilab	Physicist
Joel	Butler	Fermilab	Physicist
Erik	Gottschalk	Fermilab	Physicist
Jim	Kowalkowski	Fermilab	Engineer
Margaret	Votava	Fermilab	Engineer
John	Cassel	Illinois	Graduate Student
Michael	Haney	Illinois	Engineer
Ravi	Iyer	Illinois	Professor
Zbigniew	Kalbarczyk	Illinois	Research Scientist
Qilun	Liu	Illinois	Graduate Student
Mats	Selen	Illinois	Professor
Keith	Whisnant	Illinois	Graduate Student
John	Gross	Pittsburgh	Graduate Student
Daniel	Mosse	Pittsburgh	Professor
Leland	Hovey	Syracuse	Graduate Student
Jae	Oh	Syracuse	Professor
Sheldon	Stone	Syracuse	Professor

Madhura	Tamhankar	Syracuse	Graduate Student
Ted	Bapty	Vanderbilt	Research Professor
Sandeep	Neema	Vanderbilt	Research Scientist
Steve	Nordstrom	Vanderbilt	Graduate Student
Jason	Overdorf	Vanderbilt	Graduate Student
Paul	Sheldon	Vanderbilt	Professor
Shweta	Shetty	Vanderbilt	Graduate Student
Eric	Vaandering	Vanderbilt	Postdoc
Divya	Vashishtha	Vanderbilt	Graduate Student